IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

DATA ENGINE TECHNOLOGIES LLC,

Plaintiff,

v.

GOOGLE INC.,

Defendant.

C.A. No. 14-1115-LPS-CJB

GOOGLE INC.'S INITIAL CLAIM CONSTRUCTION BRIEF

OF COUNSEL:

Jonathan K. Waldrop
Darcy L. Jones
Robert P. Watkins III
Marcus A. Barber
John Downing
Heather S. Kim
KASOWITZ, BENSON, TORRES & FRIEDMAN LLP
333 Twin Dolphin Drive, Suite 200
Redwood Shores, California 94065
(650) 453-5170

Jeffrey J. Toney
KASOWITZ, BENSON, TORRES & FRIEDMAN LLP
1349 West Peachtree Street N.W., Suite 1500
Atlanta, Georgia 30309
(404) 260-6080

Dated: November 3, 2015

Frederick L. Cottrell, III (#2555)
Jason J. Rawnsley (#5379)
RICHARDS, LAYTON & FINGER, P.A.
920 N. King Street
Wilmington, Delaware 19801
(302) 651-7700
cottrell@rlf.com
rawnsley@rlf.com

Attorneys for Defendant
GOOGLE INC.

TABLE OF CONTENTS

Page(s)

TABLE OF CONTENTS
(continued)

Page(s)

TABLE OF AUTHORITIES

Page(s)

Cases

TABLE OF AUTHORITIES
(continued)

Page(s)

I. INTRODUCTION

Pursuant to the Scheduling Order (D.I. 16), and the Stipulation and Order Amending Claim Construction Deadlines (D.I. 49), Defendant Google Inc. ("Google") respectfully submits this opening claim construction brief directed to the construction of the disputed claim terms in the five patents asserted by Data Engine Technologies LLC ("DET"), U.S. Patent Nos. 5,303,146 ("the '146 patent"), 5,590,259 ("the '259 patent"), 5,784,545 ("the '545 patent"), 5,282,551 ("the '551 patent"), and 5,623,591 ("the '591 patent"). Each of these patents relates to spreadsheet functionality.

There are currently 11 claim construction disputes among the 5 asserted patents. Google's proposed constructions harmonize the claims with the definitions and characterizations of the inventions in the specifications, and with the patentees' statements to the U.S. Patent & Trademark Office ("PTO") during prosecution. DET's reliance on "plain and ordinary meaning" for most proposed constructions, however, would fail to aid the jury, and would effectively leave claim construction in the hands of experts and other witnesses and not the Court.

II. LEVEL OF ORDINARY SKILL IN THE ART

When construing claim terms, the Court should do so from the vantage point of a person of ordinary skill in the art at the time the application was filed. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1313 (Fed. Cir. 2005). As detailed below, the first application to which the '259, '545, '551, and '591 patents claim priority was filed on April 8, 1992. The first application to which the '146 patent claims priority was filed on March 11, 1993. A person of ordinary skill in the art of the field of the alleged inventions would have had an undergraduate degree in Computer Science, or equivalent course work, and two or more years' experience working as a programmer or software user interface designer with general familiarity with databases and/or spreadsheet-style applications.

III. BACKGROUND

A. Overview of the Asserted Patents

The five asserted patents disclose spreadsheet user interface and data entry functionality, as follows:

- ’146 patent (“Scenario Management” patent): The ’146 patent is directed to methods for specifying an area of cells within a spreadsheet to be tracked by the system for modifications, and determining and identifying for a user changes between the information in an initial (e.g., “baseline”) set of cells and subsequent versions.
- ’259, ’545, and ’551 patents (collectively, the “Spreadsheet Tab Patents”): The Spreadsheet Tab Patents are directed to user-modifiable “notebook” type tabs used to identify and navigate individual sheets in a three-dimensional (“3D”) spreadsheet. A “3D spreadsheet” refers to a computer-based spreadsheet document comprised of multiple two-dimensional spreadsheets (spreadsheets with cells arranged in rows and columns). All of these patents are continuations of common patent application 07/866,658 (the ’658 application), filed on April 8, 1992.
- ’591 patent (“User Interface Builder” patent): The ’591 patent is a continuation-in-part of the ’658 application, and is directed to methods for creating user-interface components for spreadsheet-based applications.

B. The ’146 Patent (“Scenario Management” Patent)

The ’146 patent discloses an alleged improvement on prior art “scenario management” functionality in spreadsheets, by allowing a user to create and manage “best case, worst case” what-if problems. (’146 patent, 7:15-25.) As the patent explains, “[s]cenarios are a form of model analysis where a user can create various models, by changing data in a base model and seeing how the changes affect formulas based on them." (*Id.*.) In distinguishing prior art scenario management systems from the "present invention," the '146 patent explains that:

> [T]he present invention provides a more powerful and interactive approach to scenario analysis in the form of a Scenario manager. The manager automatically tracks value changes as one enters new data sets, with the added ability to allow a user at any time to name and save scenario variations in report form. In general operation the user decides how much of the notebook (e.g. a block, a page, or an entire notebook) to include in his or her scenario category. Within this "capture area," the user decides which information is to change (i.e., specifies "changing cells") with each desired scenario.

(*Id.*, 7:62-8:6.)

To create the scenario, the user must first specify the initial base set, which is used as the reference from which new scenarios are created and compared. A flowchart showing the "present invention for scenario management is shown (right) in Figure 5A." (*Id.*, 12:50-52.) <u>At step 501</u>, "the user specifies a capture area that is [] an area in which the system will track user-specified changes." (*Id.*, 12:52-54.) The specification explains that "the user decides how much of the notebook e.g., a block, a page, or an entire notebook to include in his or her scenario category." (*Id.*, 8:1-3.) Figure 5A (right) confirms that the specifying step includes selecting both the capture area and the base case by the user. (*Id.*, 3:3-6 ("In an exemplary method of the present invention, the user first specifies a capture area (such as notebook page, or block) and a baseline (i.e., the base or standard case which is to serve as a reference).") The user's specifying an initial base set is critical to the invention because the base set is used as the reference to which new scenarios are compared. (*Id.*, 9:38-44 ("The Scenario

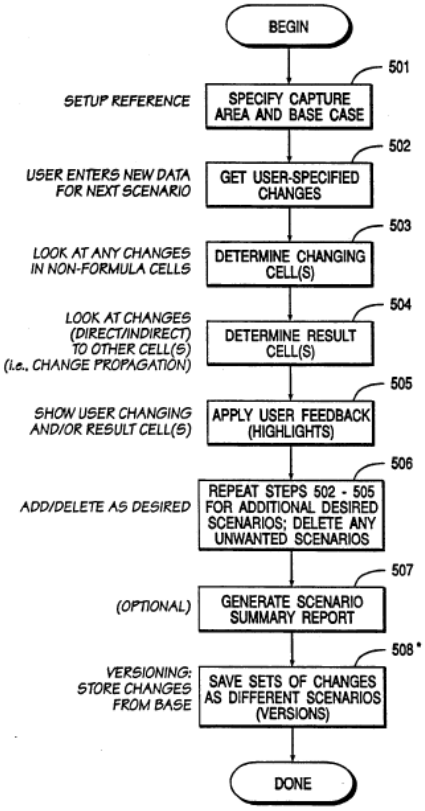
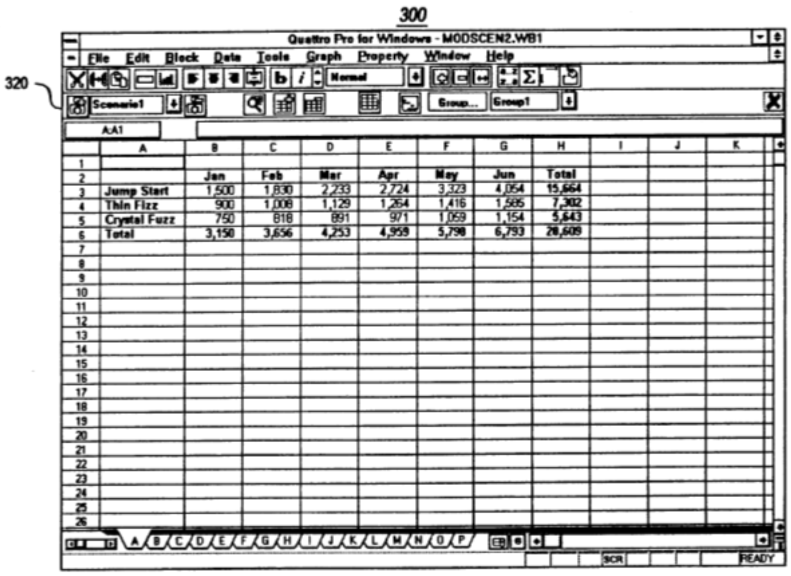

BEGIN
501
SETUP REFERENCE
SPECIFY CAPTURE AREA AND BASE CASE
502
USER ENTERS NEW DATA FOR NEXT SCENARIO
GET USER-SPECIFIED CHANGES
503
LOOK AT ANY CHANGES IN NON-FORMULA CELLS
DETERMINE CHANGING CELL(S)
504
LOOK AT CHANGES (DIRECT/INDIRECT) TO OTHER CELL(S) (i.e., CHANGE PROPAGATION)
DETERMINE RESULT CELL(S)
505
SHOW USER CHANGING AND/OR RESULT CELL(S)
APPLY USER FEEDBACK (HIGHLIGHTS)
506
ADD/DELETE AS DESIRED
REPEAT STEPS 502 - 505 FOR ADDITIONAL DESIRED SCENARIOS; DELETE ANY UNWANTED SCENARIOS
507
(OPTIONAL)
GENERATE SCENARIO SUMMARY REPORT
508
VERSIONING: STORE CHANGES FROM BASE
SAVE SETS OF CHANGES AS DIFFERENT SCENARIOS (VERSIONS)
DONE

FIG. 5A

Manager tracks changes against a baseline or base version. . . . Before various edits, assumptions, and the like are applied to the base model, the user simply instructs the system to capture the baseline. If a user is working with an empty notebook, on the other hand, he or she can enter any labels and formulas before setting the baseline.”).)

Figure 3A of the ’146 patent illustrates a base data model before a new scenario is created. (*Id.*, 3:57-60 (“FIG. 3A is a bitmap screenshot illustrating a scenario management interface of the present invention; a sample model (worksheet), which is to serve as a base case, has been loaded into the system.”).) As shown below, the base case in Figure 3A is a spreadsheet listing sales of various products over the course of six months.


300
320
Quattro Pro for Windows - MODSCEN2.WB1
File Edit Block Data Tools Graph Property Window Help
Normal
Scenario1
Group...
Group1
A:A1
A B C D E F G H I J K L M N O P
SCR
READY

	A	B	C	D	E	F	G	H
1								
2		Jan	Feb	Mar	Apr	May	Jun	Total
3	Jump Start	1,500	1,830	2,233	2,724	3,323	4,054	15,664
4	Thin Fizz	900	1,008	1,129	1,264	1,416	1,585	7,302
5	Crystal Fuzz	750	818	891	971	1,059	1,154	5,643
6	Total	3,150	3,656	4,253	4,959	5,798	6,793	28,609

FIG. 3A

At step 502, after the user selects the base set, the user proceeds to modify the base set to create new data scenarios. (*Id.*, 12:59-63 (“At step 502, user specified changes are received. Here, the user edits desired cells (typically entering new cell values) for creating a new scenario.”), 8:3-6 (“Within this ‘capture area, the user decides which information is to change . . . with each desired scenario.”); 3:28-32 (“After creating desired scenarios, the user may instruct the system to save these new versions, together with the baseline, on a storage device.”).) Once the user has completed the desired modifications, the user captures the new scenario, by clicking a button. (*Id.*, 10:40-43 (“the user invokes the Capture Scenario button 321 to capture the instant version as a new scenario.”); 8:44-47 (“Identify Scenario button 324 captures the current scenario data; by clicking Add Scenario button 321, one may associate this snapshot with a particular scenario name.”) Figure 5B illustrates the distinction between the base and various new scenarios, also referred to in Figure 5B as “version” 1, 2, and 3.

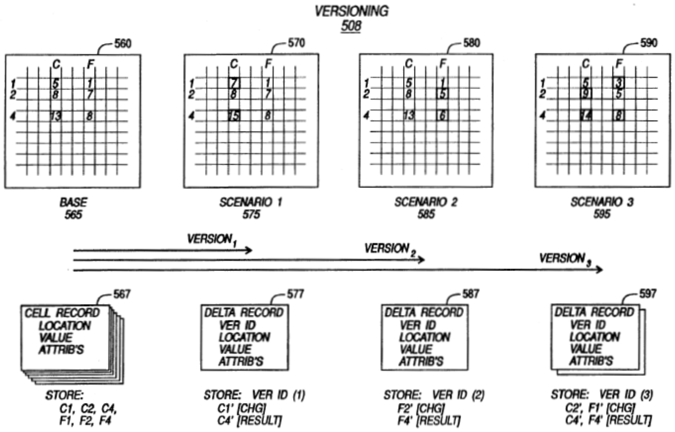


FIG. 5B

Figure 5B demonstrates that each new scenario (also referred to in Figure 5B as versions) is created by making changes to the “Base.” At steps 503 and 504, the system locates any changes in the non-formula and formula cells when compared against corresponding ones in the base set. (*Id.*, 10:1-5 (“Upon invocation of the Identify Scenario button 324, the system identifies the current scenario, that is, compares the data of the instant case with that of the base version; any cells which change from the user’s entries are automatically identified.”).) At step 508, the ’146 patent discloses storing the changes or “delta records” shown in Figure 5B. (*Id.*,13:19-21 (“As shown by the figure[5b], the different versions or scenarios represent incremental changes from the base.”).)

C. <u>The ’259, ’545, and ’551 Patents (“Spreadsheet Tab” Patents)</u>

The ’259, ’545, and ’551 patents are directed to the use of notebook tabs to access and reference individual sheets in three-dimensional (3D) spreadsheet software. (’259, ’551, ’545 patent, Abstract (“displaying a plurality of page identifiers for selecting individual pages, and further including a preferred syntax for referencing information.”) The common specification acknowledges that numerical analyses, particularly financial ones, were originally performed by hand on accountants’ columnar pads with a pencil and calculator. (’259 patent, 1:25-36.) The patents explain that, with the advent of computers, the columnar pad, pencil, and calculator were replaced with spreadsheet software programs on computers that could hold more information than their paper counterparts and easily recalculate cells. (*Id*., 1:37-2:24.) The specification explains that three-dimensional (3D) spreadsheet software in existence prior to the patents allowed a user to consolidate multiple spreadsheets into a single worksheet consisting of multiple spreadsheets. (*Id*., 2:66-3:24.)

The ’259, ’545, and ’551 patents set forth an alleged improvement over prior art 3D spreadsheets: employing page identifiers displayed as tabs similar to “tabs from an ordinary paper notebook” to identify individual sheets. (’259 patent, Abstract; 2:27-4:5; 8:10-33.) The notebook tabs (members 261a, 262a, and 263a) are illustrated below in Figure 2D:

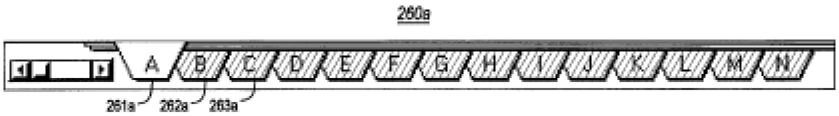
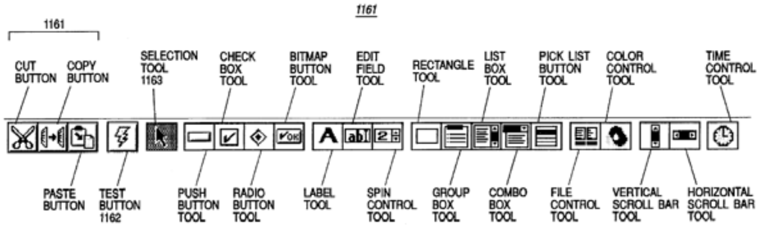


FIG. 2D

Each asserted claim in the ’259, ’545, and ’551 patents is expressly directed to a method of using a computer based electronic spreadsheet with multiple spreadsheet pages, with each individual spreadsheet page containing an associated user-customizable spreadsheet page identifier.

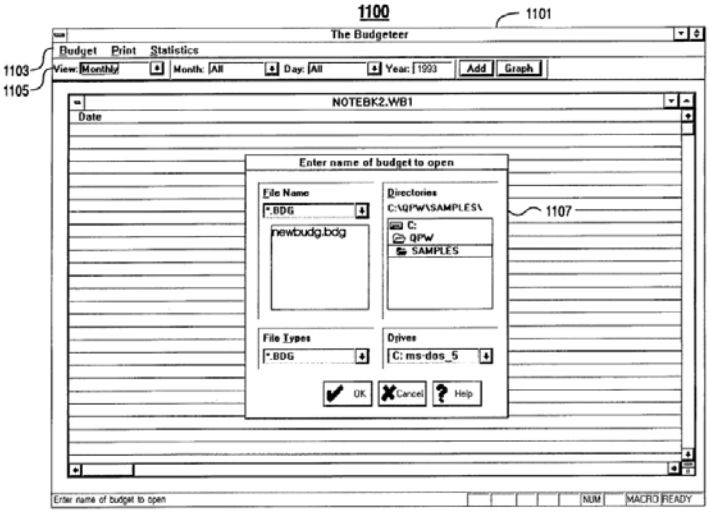
D. The ’591 Patent (“User Interface Builder” Patent)

The ’591 patent shares much of its specification with the Spreadsheet Tab Patents (’259, ’551, and ’545), and contains additional disclosures that support its claims. The ’591 patent’s claims are directed to a user interface builder (“UI Builder”) feature in an electronic spreadsheet application, which consists of tools to build custom applications (end user-facing programs) to access and modify spreadsheet data. (’591 patent, Abstract (“The system includes a spreadsheet application development module having a user interface (UI) builder.”); 25:57-26:46.) The patent discloses a toolbar of available user interface objects (text boxes, drop-down menus, and buttons) which a developer or end-user can select and add to an empty dialog window to create applications. (’591 patent, 28:27-35) These user interface objects are presented in a “dialog window toolbar.” *Id.* A snapshot illustrating the toolbar is shown below.



(’591 patent, Fig. 11H; *see also*, 5:62-65 (“FIG. 11H is a bitmap screenshot illustrating a toolbar of the system of the present invention for use in the dialog window.”))

The user interface objects are then “linked” to cells such that any change made to the user interface object is propagated to the cell, and vice versa (in other words, the linking is bi-directional). (*Id.*, 46:19-29 (“a plurality of objects may be connected so that a change in the property of one of the objects will automatically propagate to the corresponding properties of other objects.”)) As an example of a spreadsheet application built using the invention, the patent describes a “Budgeteer” application consisting of various drop-down menus and buttons, which allow an end user to enter information to create and save new budgets, and review statistics. ('591 patent, 26:53-27:40; Fig. 11A)



View: Monthly
Month: All
Day: All
Year: 1993

File Name: *.BDG
Directories: C:\QPW\SAMPLES\
File Types: *.BDG
Drives: C: ms-dos_5

Enter name of budget to open

FIG. 11A

('591 patent, Fig. 11A)

IV. LEGAL STANDARDS FOR CLAIM CONSTRUCTION

Claim construction is a matter of law for the court. *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995) (*en banc*), *aff'd,* 517 U.S. 370 (1996). "It is a bedrock principle of patent law that the claims of a patent define the invention to which the patentee is entitled the right to exclude." *Phillips,* 415 F.3d at 1312 (internal quotations and citations omitted). The correct claim construction "stays true to the claim language and most naturally aligns with the patent's description of the invention." *Renishaw PLC v. Marposs Societa' per Azioni,* 158 F.3d 1243, 1250 (Fed. Cir. 1998).

As the Court discussed in *Phillips*, several tenets undergird a proper claim construction analysis. Principal among them is that "the construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction." *Philips,* 415 F.3d at 1316. But in doing so, "[c]laims must be 'interpreted with an eye toward giving effect to all terms in the claim.'" *Becton, Dickinson & Co. v. Tyco Healthcare Group, LP*, 616 F.3d 1249, 1257 (Fed. Cir. 2010) (quoting *Bicon, Inc. v. Straumann Co.*, 441 F.3d 945, 950 (Fed. Cir. 2006)). Therefore, claim terms generally should not be construed in a manner that renders other claim terms surplusage. *Tex. Instruments Inc. v. ITC*, 988 F.2d 1165, 1171 (Fed. Cir. 1993).

A second canon in claim construction analysis is that "the specification 'is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.'" *Phillips*, 415 F.3d at 1315 (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)). *Phillips* made clear that a person of ordinary skill in the art is deemed to read the claim term not only in the context of a particular claim in which the disputed term appears, but in the context of the entire patent, including the specification. *Id.* at 1313. Additionally, as intrinsic evidence, the prosecution history plays an important role in claim interpretation. It helps demonstrate how the inventor and the PTO understood the patent. *Id.* at 1317.

V. GOOGLE'S CONSTRUCTIONS SHOULD BE ADOPTED

A. The '146 Patent ("Scenario Management" Patent) Disputed Terms

1. "Specifying a base set of information cells"

Claim Term(s)	Google's Proposed Construction	DET's Proposed Construction
"Specifying a base set of information cells" Claim: 1	User selecting a set of cells in an open notebook as a base set from which user defined scenarios are created.	Plain and ordinary meaning. No construction necessary.

a. Google's Proposed Construction Consistently Defines the Limitation According to the Patent Specification.

The '146 patent repeatedly and consistently uses the term "base" to refer to a set of cells used as a reference to create scenarios. As explained in detail in the Background section of the specification, the "user specifies a capture area (such as notebook page, or block) ***and*** a baseline (i.e., ***the base or standard case*** which is to serve as a reference)." ('146 patent, 3:3-6 (emphasis added).) Figure 5A, block 501, additionally confirms that the specifying step includes specifying the "base case."

The user's specifying a base set is critical to the invention because it is the "base set" that will be used as a reference from which new scenarios will be created and from which the comparison is made. (*Id.*, 9:38-44 ("The Scenario Manager tracks changes against a baseline or base version.").) Figure 5B (below) also uses the term "Base," and the patent likewise states that "spreadsheet model 560 is specified as base 565." (*Id.*, 13:22-23).)

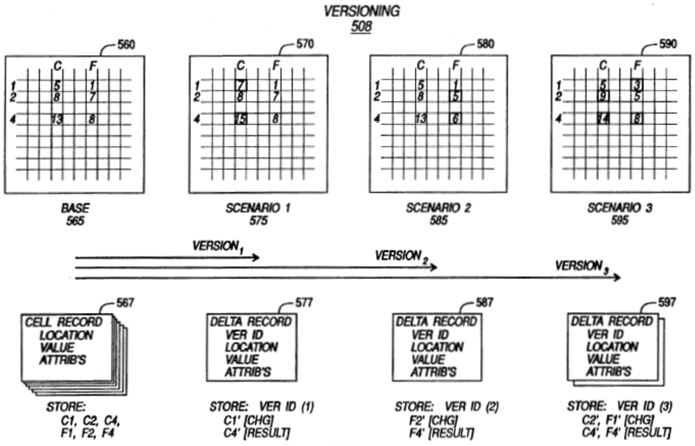


FIG. 5B

Scenarios 1, 2, and 3 in Figure 5B are created by making modifications from "Base" 565 and the changes are determined by "comparing cells in each new version against corresponding ones in the base set." (*Id.*, 13:15-18 ("At step 508, the user saves the current notebook with the different scenarios (versions) being stored as changes from the parent or ***base***.").)

The '146 patent's description of the term "base" comports with Google's proposed definition of "specifying a base set of information cells" *i.e.*, user selecting a set of cells in an open notebook as a base set (565) from which user defined scenarios are created from and compared against (570, 580, and 590). *See Phillips*, 415 F.3d at 1317 ("It is therefore entirely appropriate for a court, when conducting claim construction, to rely heavily on the written description for guidance as to the meaning of the claims.")

b. Google's Proposed Construction is Necessary Because DET Seeks to Read Out the "Specifying" Limitation.

DET's "plain and ordinary meaning" proposal seeks to read out the entire specifying limitation. DET argues that, in spreadsheet software such as Google Sheets that tracks changes to all cells, the limitation of "specifying a base set of information cells for the system to track changes" is met when a user simply opens a spreadsheet (thereby implicitly specifying the entire blank workbook). *Flex-Rest, LLC v. Steelcase, Inc.*, 455 F.3d 1351, 1361 (Fed. Cir. 2006) ("Flex-Rest's attempt to avoid the sidewall limitation by characterizing it as 'inconsequential' disregards the basic patent law doctrine that every limitation of a claim is material."). DET's position is inconsistent with the usage of the "specifying" term throughout the '146 patent, which consistently explains providing the user with user interface tools for specifying the base set when the workbook is open. (*See*, *e.g.*, '146 patent, 9:48-49 ("the user invoking the Capture Scenario button 321.")) Indeed, the patent explains that "when a user is working with an empty notebook … he or she can enter any labels and formulas ***before*** setting the baseline." (*Id.*, 9:44-47), *see also ERBE Elektromedizin GmbH v. Int'l Trade Comm'n*, 566 F.3d 1028, 1034 (Fed. Cir. 2009) ("We generally do not construe claim language to be inconsistent with the clear language of the specification"; "[u]sually, it is dispositive.").) The '146 specification makes clear that the spreadsheet program is open and the "capture" tools are available before the user can specify the base. DET's position also conflicts with the claim preamble that provides a "spreadsheet system for modeling ***user-specified information***." *Bell Commc'ns Research, Inc. v. Vitalink Commc'ns Corp.*, 55 F.3d 615, 620 (Fed. Cir. 1995) ("[A] claim preamble has the import that the claim as a whole suggests for it. In other words, when the claim drafter chooses to use both the preamble and the body to define the subject matter of the claimed invention, the invention so defined, and not some other, is the one the patent protects.").

Google's construction should therefore be accepted to prevent DET's argument that the limitation "specifying a base set of information cells" is met when nothing is specified.

c. Google's Proposed Construction is Consistent with Dependent Claims that Also Use the Term "Base Set"

Google's proposed construction is further consistent with limitations (b) and (c) in claim 1 which also include the limitation "base set." ('146 patent, Claim 1 ("(b) creating a new version of the data model by modifying at least one information cell from the specified base set; and (c) automatically determining cells of the data model which have changed by comparing cells in the new version against corresponding ones in the base set.").) Even if the specifying limitation could be met by simply opening a new spreadsheet, one would assume that the remaining uses of the term "base set" should also mean a new blank spreadsheet. *Am. Piledriving Equip., Inc. v. Geoquip, Inc.*, 637 F.3d 1324, 1333 (Fed. Cir. 2011) ("Where a claim term is used consistently throughout the claims, the usage of [the] term in one claim can often illuminate the meaning of the same term in other claims."). DET's interpretation of the limitation, however, does not require "automatically determining cells of the data model which have changed by comparing cells in the new version against corresponding ones in the ***base set*** [new blank spreadsheet]."

('146 patent, Claim 1.) Because Google's proposed construction is consistent with the claim language and intrinsic record, Google respectfully requests that the Court adopt its construction.

2. "Version" terms: "Different versions;" "New version;" and "Base version"

Claim Term(s)	Google's Proposed Construction	DET's Proposed Construction
"Different versions" Claims: 1 and 26.	User defined scenarios made by modifying the base version.	Plain and ordinary meaning. No construction necessary.
"New version" Claims: 1, 26, 27, 32, 33, and 34.	New user defined scenario made by modifying the base version.	Plain and ordinary meaning. No construction necessary.
"Base version" Claims: 1 and 26.	A single user defined reference version which is modified to create a new scenario.	Plain and ordinary meaning. No construction necessary.

Google seeks constructions of the terms "different versions," "new version," and "base version" that demonstrate that the user, and not the system, creates and defines new or different versions, and that all versions other than the base version are created by modifying the base version. Google's proposed construction is supported once again by the preamble of claim 1 ("modeling user-specified information in a data model"), as well as the stated purpose of the "present invention," which is to demonstrate user-specified changes over a base version:

> [T]he present invention provides a more powerful and interactive approach to scenario analysis in the form of a Scenario manager. The manager automatically tracks value changes ***as one enters new data sets, with the added ability to allow a user at any time to name and save scenario variations in report form***.

(*Id.*, 7:62-68 (emphasis added).) In its recent response to Google's Motion to Strike, DET asserted that Google Sheets itself modifies the base set to create new versions. These assertions, when compared to the '146 patent specification, illustrate the need for Google's proposed construction (and belie DET's proposed "plain and ordinary meaning" construction):

DET's Position	'146 Patent Specification
Google Sheets itself ***performs the modification***. The user of any spreadsheet system cannot reach into the source code or data structures of an application and change the value of any cell directly. Instead the user inputs a value into Google's computer program, the program then processes that input and translates it into modifying the data structures underlying the spreadsheet user interface. (DET Infringement Contentions)	"Next, ***the user proceeds to modify*** the worksheet, for example, changing the various assumptions (e.g., tax rate) employed ***to create*** the model." ('146 patent, Col. 3:6-9.)

Nowhere does the specification of the '146 patent describe that a system (as opposed to a user) modifies a base version to create and define new or different versions. *Phillips*, 415 F.3d at 1316 ("The construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction.") Because Google's proposed constructions naturally align with the intrinsic record, Google respectfully requests that the Court construe the claims accordingly.

3. "Maintaining the new version by storing additional information for only those portions determined to have changed"

Claim Term(s)	Google's Proposed Construction	DET's Proposed Construction
"Maintaining the new version by storing additional information for only those portions determined to have changed"	Maintaining the new version by storing only portions of the new version which have changed when compared against the base version.	Plain and ordinary meaning. No construction necessary.

As explained above with reference to Figure 5B, "different versions or scenarios represent incremental changes from the base." (*Id.* at 13:19-21.) When describing the storing aspect of the present invention, the '146 specification explains that the "user saves the current notebook with the different scenarios (versions) being stored as changes from the parent or base." (*Id.*, 13:15-18; *see also*, 13:24-27 ("The present invention recognizes … that each additional version [*i.e.* scenario] may be represented by just storing the difference (delta) record(s).") Google's proposed construction is directly in line with the description of the storage method described in the '146 specification. Because Google's proposed construction is consistent with the claim language and intrinsic record, Google respectfully requests that the Court adopt its construction.

B. <u>'545 Patent ("Spreadsheet Tab" Patent) Disputed Terms</u>

1. <u>"Single disk file"</u>

Claim Term(s)	Google's Proposed Construction	DET's Proposed Construction
Single disk file Claims: 1 and 35.	Stored in a file on a single physical disk.	Plain and ordinary meaning. No construction necessary.

The claims and specification of the '545 patent make clear that "single disk file" means "stored in a file on a single physical disk," in accordance with Google's proposal. DET's assertion that the term should be given its "plain and ordinary meaning," without actually identifying what that meaning is, represents nothing more than an empty statement of law that is unhelpful to the trier of fact. *See O2 Micro Int'l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1360 (Fed. Cir. 2008) ("When the parties raise an actual dispute regarding the proper scope of these claims, the court, not the jury, must resolve that dispute.").

Google's proposed construction of "single disk file" as "stored in a file on a single physical disk" comports with the patent's use of the term. Claims 1 and 35 both recite the step of storing a plurality of spreadsheet pages in a single disk file. To make clear that spreadsheet pages are saved as a "single disk file" on a single physical disk, the specification explains that the "single disk file" is stored in the "mass storage" of the system. ('545 patent, 5:27-33.) For example, the '545 patent discloses only one embodiment, shown pictorially in Figure 1A below, which comports with Google's proposed construction requiring a disk file to be stored in "mass storage." *See Johns Hopkins Univ. v. CellPro, Inc.*, 152 F.3d 1342, 1355 (Fed. Cir. 1998) (stating that “the only disclosed embodiment of the claimed [invention] is highly indicative of the scope of the claims”).

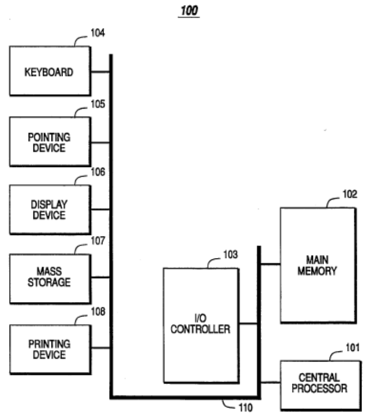


FIG. 1A

Figure 1A is described as a system comprising general hardware components of a generic computer including “a central processor 101, a main memory 102, … and *a mass storage* 107 (e.g., hard disk).” (’545 patent, 5:27-33.) The specification further states that “the notebook 250 includes 256 spreadsheet pages and one Graphs page, *all of which are saved as a single disk file on the mass storage* 107.” (*Id*., 7:59-62 (emphasis added). The specification’s description of “mass storage” is consistent with its common usage in the field of personal computers, which defines the term as “large-capacity backup storage such as a hard disk, external hard disk, cartridge, or streaming tape.” *IBM Dictionary of Computing*, 10th Ed. (1994) at 420. Further, the “hard disk” is defined as “a rigid magnetic disk such as the internal disks used in the system units of personal computers and in external hard disk drives.” *Id*.

Accordingly, the specification and common usage of the term make clear that spreadsheet pages are saved as a “single disk file” on a hardware component of the disclosed computer system, which is consistent with Google’s proposed construction, “stored in a file on a single physical disk.” Google respectfully requests that the Court adopt its construction.

C. ’551 patent (“Spreadsheet Tab” Patent) Disputed Terms

1. “Storing said first and second pages of the plurality of cell matrices such that they appear to the user as being stored within a single file”

Claim Term(s)	Google’s Proposed Construction	DET’s Proposed Construction
Storing said first and second pages of the plurality of cell matrices such that they appear to the user as being stored within a single file Claims: 1.	Indefinite.	Storing said first and second pages of the plurality of cell matrices such that they are accessible to the user by a single file name.

The phrase “storing said first and second pages of the plurality of cell matrices such that they appear to the user as being stored within a single file” is indefinite and remains indefinite even under DET’s proposed construction. According to statute, the specification of a patent shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the inventor or a joint inventor regards as the invention. 35 U.S.C. § 112, ¶ 2 (2012). As the Federal Circuit explained in *Datamize, LLC v. Plumtree Software, Inc.*, when it held that “aesthetically pleasing” was indefinite despite having a clear meaning, purely subjective claim terms that depend on the opinion of a particular user cannot satisfy the requirements of Section 112. 417 F.3d 1342, 1347-50 (Fed. Cir. 2005), *abrogated on other grounds by Nautilus, Inc. v. Biosig Instruments, Inc.*, 134 S. Ct. 2120 (2014). Indeed, in the absence of an objective standard, claim scope is indefinite where the determination of whether an accused product falls within or outside of the scope of the claim depends completely on the unrestrained opinion of a particular individual.

The requirement that the first and second pages of the plurality of cell matrices “appear” to the user as being stored within a single file in claim 1 of the ’551 patent is entirely subjective on its face, depending directly on whether the user perceives that the matrices have been stored within a single file. This limitation is indefinite because it would "depend solely on the unrestrained, subjective opinion of a particular individual purportedly practicing the invention." *Datamize*, 417 F.3d at 1350-51.

When a subjective term is used in a claim, "the court must determine whether the patent's specification supplies some standard for measuring the scope of the [term]." *Datamize*, 417 F.3d at 1351; *accord Interval Licensing LLC v. AOL, Inc.*, 766 F.3d 1364, 1371 (Fed. Cir. 2014) (*citing Datamize*, 417 F.3d at 1351), *cert. denied*, 2015 WL 2338493 (U.S. Oct. 5, 2015). The specification and prosecution history of the '551 patent fail to provide an objective standard defining what a user must observe in order for it to "appear" that the matrices have been stored within a single file. The specification merely states that the notebook interface of the present invention provides a "means for organizing many spreadsheets together into one file" and that a disk file can have the same name as the title of a notebook. ('551 patent, 11:26-47; 15:30-54) Without an objective standard, the term is indefinite. *See Interval Licensing*, 766 F.3d at 1370-71; *Datamize*, 417 F.3d at 1350-51, 1356 ("The [] patent . . . fails to provide any objective way to determine whether the look and feel of an interface screen is 'aesthetically pleasing.'"); *Prolifiq Software, Inc. v. Veeva Sys. Inc*, No. 13-03644, 2014 WL 3870016, at *8 n.6 (N.D. Cal. Aug. 6, 2014) ("[W]here a claim term allows the scope of the invention to be determined by the unrestrained, subjective opinion of the person practicing the invention, then no one, including a person skilled in the art, can determine with reasonable clarity the scope of the invention.").

DET proposes that the phrase be construed to mean "storing said first and second pages of the plurality of cell matrices such that they are accessible to the user by a single file name." However, the addition of "such that they are accessible to the user by a single file name" merely attempts to equate "appear" with "accessible" but provides no objective standard for how the matrices must "appear." Regardless, the Supreme Court explained that a patent does not satisfy the definiteness requirement of § 112 merely because "a court can ascribe *some* meaning to a patent's claims." *Nautilus*, 134 S.Ct. at 2130. The claims, when read in light of the specification and the prosecution history, must provide objective boundaries for those of skill in the art. *See id.* at 2130 & n.8 (indicating that there is an indefiniteness problem if the claim language "might mean several different things and 'no informed and confident choice is available among the contending definitions'"); *see also Datamiz*e, 417 F.3d at 1350-51 ("[E]ven if we adopted a completely subjective construction . . . this would still render the [] patent invalid.") Additionally, in *Halliburton Energy Services, Inc. v. M-I LLC*, the Federal Circuit specifically stated that "[t]he fact that [the patent holder] can articulate a definition supported by the specification . . . does not end the inquiry. Even if a claim term's definition can be reduced to words, the claim is still indefinite if a person of ordinary skill in the art cannot translate the definition into meaningfully precise claim scope." 514 F.3d 1244, 1251 (Fed. Cir. 2008) (finding the term "fragile gel" indefinite); *see also Union Pac. Res. Co. v. Chesapeake Energy Corp.*, 236 F.3d 684, 692 (Fed. Cir. 2001) (finding the term "comparing" indefinite when it "could undoubtedly have other meanings to a person of skill in the art."). And as Dr. John D. Kubiatowicz recognized in the attached declaration in support of this brief, the specification and the prosecution history of the '551 patent fails to provide objective boundaries for a person of skill in the art to determine what a user must observe in order for it to "appear" that the matrices have been stored within a single file. (Declaration of Dr. John D. Kubiatowicz in support of Google Inc.'s Initial Claim Construction Brief at ¶¶ 11-14)

Because the scope of the claim cannot depend on the subjective opinion of the user, claim 1 of the ’551 patent is indefinite under Section 112. Google respectfully requests that the Court adopt its construction that the term is indefinite.

D. ’591 patent (User interface object creation) Disputed Terms

1. “Generating a User Interface Object” terms:
“In response to first user input, generating a user interface object;”
“In response to third user input, generating a user interface object” and
“Receiving first user input for generating a user interface control of a plurality of different types”

Claim Term(s)	Google’s Proposed Construction	DET’s Proposed Construction
“In response to first user input, generating a user interface object of a predefined type distinct from cell objects” Claims: 1.	Generating a user interface object of a predefined type distinct from cell objects in response to a user selecting a user interface object type in a graphical user interface.	Plain and ordinary meaning. No construction necessary.
“In response to third user input, generating an additional user interface object of a predefined additional type distinct from cell objects” Claims: 3.	Generating an additional user interface object of an additional predefined type distinct from cell objects in response to a user selecting a user interface object type in a graphical user interface.	Plain and ordinary meaning. No construction necessary.
“Receiving first user input for generating a user interface control of a plurality of different types” Claims: 13.	Receiving user input selecting a user interface control type in a graphical user interface.	Plain and ordinary meaning. No construction necessary.

a. The ’591 Patent Consistently Explains that the “user input [for] generating user interface objects” is Choosing Objects from a Graphical User Interface such as a Toolbar

As discussed above, the claims of the ’591 patent are directed to a “UI Builder” feature added to electronic spreadsheet software to allow a user to easily assemble user-friendly applications that use the underlying spreadsheet to organize and operate on information. The dispute over the above terms concerns whether the claims broadly cover any means of generating user interface objects to build a spreadsheet application (DET's position), or if instead the "user input [for] generating user interface objects," according to the teachings of the patent, consists more specifically of selecting visual representations of user interface object types from a graphical user interface such as a toolbar or similar menu (Google's position).

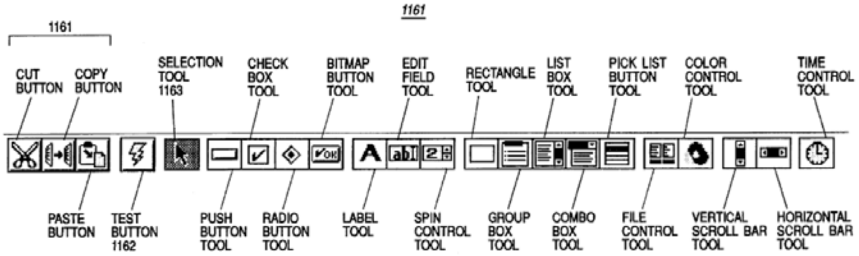
The specification and prosecution history clearly demonstrate that building spreadsheet applications by selecting user interface object types from a graphical user interface such as a toolbar was the goal of the invention. The patent explains that "[t]he system of the present invention includes a dialog window toolbar" which is used to design custom applications:

> The system of the present invention includes a dialog window toolbar 1160, shown in FIG. 11H, which contains tools that are specifically designed to help the developer build dialog boxes and toolbars. It includes tools which let the developer quickly create, copy, move and test controls. Controls are elements the developer puts in a dialog box to gather information from the user or to perform a desired action.

('591 Patent, 29:31-38.) The Federal Circuit has repeatedly recognized that it is proper to rely on inventors' descriptions of "the invention" or "the present invention" to determine the scope of the claimed invention. *See, e.g.*, *Trading Techs. Int'l, Inc. v. eSpeed, Inc.*, 595 F.3d 1340, 1353-54 (Fed. Cir. 2010) (specification's reference to a "one click centering feature" as part of "the present invention" "strongly suggest[ed] that the claimed re-centering command require[d] a manual input, specifically, a mouse click"); *TiVo, Inc. v. EchoStar Commc'ns Corp.*, 516 F.3d 1290, 1300 (Fed. Cir. 2008) (holding that where physical separation functionality was described as an aspect of "the invention," and embodiments in specification were consistently described with this functionality, physical separation was a claim limitation); *Verizon Servs. Corp. v. Vonage Holdings Corp.*, 503 F.3d 1295, 1308 (Fed. Cir. 2007). Here, the patent clearly explains that in "<u>the system of the present invention</u>" the "user input" for generating user interface objects (also called “user interface controls”) is selecting an object from a toolbar with a variety of objects/controls that a user can “quickly create, copy, move, and test.”

The ’591 patent then provides a number of examples of building applications according to the invention. In each example, the patent explains that once the “UI Builder” feature is started (by selecting “Tools,” then “UI Builder”), the user generates user interface objects by selecting buttons corresponding those user interface objects from a toolbar. *See, e.g.*:

- ’591 patent, Fig. 11H, described as “a bitmap screenshot illustrating a toolbar of the system of the present invention for use in the dialog window.” *Id.*, 5:63-65 (emphasis added).



- *Id.*, 35:4-52: “Here is an example of grouping controls together: First, a developer chooses Tools | UI Builder to display a new dialog window. Next, he or she uses the Rectangle tool to create a rectangle that is big enough to contain the OK button. Next he or she moves the OK button into the rectangle.” [*See,* Fig. 11H above showing “Rectangle tool,” and “Bitmap Button tool” button with “OK” on the UI Builder’s toolbar.]

- *Id.*, 41:29-34: “The following example shows how to create a scroll bar that changes the active notebook’s zoom factor, using a SEND link command. First, the developer should create a new dialog window, then create a vertical scroll bar control. He or she can drag a bottom handle of the scroll bar to make it longer.” [*See,* Fig. 11H above showing “Vertical Scroll Bar tool” and “Horizontal Scroll Bar tool” buttons on the UI Builder’s toolbar.]

- *Id.*, 39:40-52: “A time control shows the user the current time. A developer can use time controls to display the current time in a dialog box or toolbar. To add a clock, one first creates a time control in the dialog window. Then, he or she can right-click the time control and set Show Time to Yes.” [*See,* Fig. 11H above showing “Time Control tool” button on the UI Builder’s toolbar.]

- *Id.*, 42:45-59: “First, the developer chooses Tools | UI Builder to create a new dialog window. Then he or she chooses the Edit Field tool and click in the dialog window to create an edit field; then he or she can drag a right handle of the edit field to make it wider.” [*See,* Fig. 11H above showing “Edit Field tool” button on the UI Builder’s toolbar.]

- *Id.*, 44:57-45:5: The following example shows how to create a color control that initially gets its value from an edit field. While the dialog box appears, changing the color control's setting also sends the color control's new value to the edit field. First, the developer chooses Tools | UI Builder to create a new dialog box. Then he or she can drag the lower right corner of the window to enlarge it. Next, <u>he or she may use the Color Control tool to create a color control</u> in the upper left corner of the window, then <u>use the Edit Field tool to create an edit field</u> next to the color control, and resize the edit field so it is wider. [*See*, Fig. 11H above showing "Color Control tool" and "Edit Field tool" buttons on the UI Builder's toolbar.]
- *Id.*, 45:17-39: "The following example shows how to create a file control that gets its value from an edit field. First, the developer chooses Tools | UI Builder to display a new dialog window; it may be enlarged by dragging the lower right corner of the window. Next, <u>the developer selects a File Control tool from the dialog Toolbar</u> and clicks in the upper left corner of the window to create the file control. <u>To create an edit field, the developer chooses the Edit Field tool from the Toolbar</u> and clicks in the dialog window to the right of the file control." [*See*, Fig. 11H above showing "Color Control tool" and "Edit Field tool" buttons on the UI Builder's toolbar.]

The '591 patent consistently explains that a user generates user interface objects for applications by selecting them from a graphical user interface such as a toolbar. Google's construction properly reflects this and would aid the jury in understanding the invention.

b. "Generating user interface objects" by Other Means, Under DET's Construction, is Not Supported by the Intrinsic Record

DET may argue that the '591 patent discloses the use of macros, and that these "macros" disclosures in the patent support a broader definition of "user input [for] generating a user interface object," as required by the claims. Any such reading of the patent is misguided. The use of "macros" in the patent is associated with a different, distinct step of the claims – the step of linking a user interface object to one or more cells (*See,* '591 Patent, Claim 1 ("in response to <u>second</u> user input, linking said value property of the given cell object"); *Id.,* Claim 13 ("receiving <u>second</u> user input for linking the user interface control, <u>so generated</u>, to at least one desired information cell"))

Both asserted independent claims make clear that this step of receiving "user input [for] linking" a cell to the user interface object takes place subsequent to (and thus separate from) the earlier "user input [for] generating a user interface object." *Mformation Techs., Inc. v. Research in Motion Ltd.*¸ 764 F.3d 1392, 1398-99 (Fed. Cir. 2014) ("a claim 'requires an ordering of steps when the claim language, as a matter of logic or grammar, requires that the steps be performed in the order written, or the specification directly or implicitly requires' an order of steps.") (citing *TALtech Ltd. v. Esquel Apparel, Inc.*, 279 Fed. App'x 974, 978 (Fed. Cir. 2008)).

The specification similarly discloses that creating the custom user interface components and linking those components to one or more cells to assemble them into an application in "the present invention" are distinct steps:

> When building an application using the system of the present invention, the user (in a role as developer) can create custom components, including dialog boxes, toolbars, menus, and the like. <u>After creating these components</u>, the user can assemble them into an integrated application notebook which comprises all of the programming logic and user interface components for the spreadsheet application.

('591 patent, 25:61-26:45) The specification describes the use of macros only for this second "linking" step of assembling the user interface components into an application, not for "user input [for] generating user interface objects." (*Id.,* 28:20-26 ("The developer can use macros to assemble these components into an application. One can also use macros to automate system operations, change the system's appearance, and display dialog boxes.")).

c. Generating user interface objects by Other Means, per DET's Construction, Was Prior Art

Consistent with the above, the patentee explained during prosecution of the '591 patent that he did not invent basic adding of user interface controls in an application, and that the '591 patent disclosed a novel way of creating user interface controls and linking them to desired cells. In a February 8, 1995 Office Action, the examiner rejected all claims, finding the claims invalid as obvious over U.S. Patent No. 5,339,410 to Kanai (assigned to IBM) ("Kanai") in combination with the SAA Common User Access Advanced Interface Design Guide ("IBM SAA"). The examiner found that Kanai disclosed "a spreadsheet application by which data values are propagated in a bidirectional manner," but failed to teach "linking a user interface object to a cell object. February 8, 1995 Non-Final Rejection of Claims, DETFH0002412. (Joint Claim Construction Chart ("JCCC") (Dkt. 53) at Exhibit F) The examiner argued that the IBM SAA taught this element via its teaching that "a 'radio button' is used to change the format of the date in a date field," and that this would have been applicable to spreadsheet software. *Id.*

To overcome this rejection, the applicant argued that he did not claim to have invented the use of user interface objects as used in the cited prior art, but instead invented (1) user-created user interface objects, that (2) can be linked to desired cells:

> Kanai contains no suggestion of user-created interface objects, distinct from cells, that can be linked to desired cells. For example, Kanai's radio button used to change date format of a date field is not an instance of a user-created interface control. While Applicant does not claim to have invented the types of control shown in IBM, Applicant does not believe that it would have been obvious to allow a user to create such user interface controls and link them to desired cells to control desired properties of the cells.

(*Id.*) Google's proposed constructions properly teach that the "user input [for generating] a user interface object" in accordance with the patent consists of selecting user interface objects in a graphical user interface such as a toolbar. Google's proposed constructions, contrary to DET's, demonstrate that the invention does not cover any and all kinds of "user input [for generating] a user interface object," such as editing script or source code; this was known in prior art such as the "radio button" appearing in the Kanai and IBM SAA software described above. Google's proposed constructions, requiring "a user choosing a user interface object in a graphical user interface,” are consistent with the clear teachings of the intrinsic record and should thus be adopted.

2. “Linking" a user interface object to a cell terms control to a given information cell”:
“linking said value property of the cell with said value property of said user interface object” (Claim 1);
“linking said display attribute property of an additional given cell object to that of said additional user interface object” (Claim 3);
“method for linking user interface controls to information cells” (Claim 13);
“linking the user interface control, so generated, to at least one desired information cell” (Claim 13); and
“linking the user interface control to a given information cell” (Claim 13)

Claim Term(s)	Google’s Proposed Construction	DET’s Proposed Construction
“Linking” Claims: 1, 3, and 13.	Bidirectionally linking a property of a cell to a property of a user interface object such that when one object’s property is changed, the other’s property will be updated automatically.	Plain and ordinary meaning. No construction necessary.

The ’591 patent and file history make clear that the “linking” that takes place between the user-created user interface objects and the cells to which they are linked is “bi-directional,” meaning that any changes to one are automatically propagated to the other. The specification explains that once a UI object is” linked” to one or more cells, changes to either the UI object or the associated cell(s) are propagated to the other:

> For connection of properties between objects, when the property of one object is set (i.e., its set property method is invoked), the system checks whether a link exists which connects the property to that of another object. In this manner, a plurality of objects may be connected so that a change in the property of one of the objects will automatically propagate to the corresponding properties of other objects.

(’591 Patent, 46:16-29; *see also*, 2:26-36 (“conventional systems lack the ability to connect or link spreadsheet cells to user interface objects, so that a cell may be updated dynamically with a change to a user interface object.”))

During prosecution of the ’591 patent, the applicant likewise explained that this “bi-directional” linking set this invention apart from prior art, which only disclosed one-way linking. On December 28, 1995, the PTO examiner issued a final rejection, finding all claims obvious over U.S. Patent No. 5,255,363 to Seyler (“Seyler”). December 28, 1995 Final Rejection, DETFH0002460-68. (JCCC at Ex. K) The examiner argued that “Seyler discloses generating user interface objects both for facilitating entry of data into spreadsheet cells and for representing data in the spreadsheet cells.” Specific to the “linking” of user interface objects to cells, the examiner cited functionality in Seyler in which “a button gadget [is used] to facilitate the updating of data values for a particular spreadsheet cell.” *Id.* at DETFH0002461-62. The examiner found that this functionality in a spreadsheet application disclosed by Seyler would render obvious the ’591 patent’s disclosures of “linking” a user interface object to a spreadsheet cell object. *Id.* To overcome this rejection, the ’591 applicant argued that use of a radio button to change a date field failed to teach the “bi-directional” linking required by the ’591 patent:

> Different aspects of the present invention go well beyond the disclosure in Seyler. One distinction is the invention’s ability to connect a property of the UI control with the corresponding property of the cell so that when either has the value of the property changed, the change propagates to the other.

(May 28, 1996 Claim Amendment. DETTFH0002501-11) (JCCC at Ex. M) The same was argued in a June 7, 1996 interview between the applicant and examiner:

> The examiner contacted applicant in order to clarify applicant’s intent regarding the bi-directional linking of properties in claim 1. Applicant confirmed that in claim 1 (lines 18-20), the ‘displaying said user interface property…’ step is to be interpreted to mean that changing the value of “said value property” in a cell will update the value of the user interface object’s ‘value property.’

(June 7, 1996 Interview Summary, DETFH0002514) (JCCC at Ex. N) The examiner cited this specific point of novelty in granting the ’591 patent:

> While the prior art teaches using a gadget to represent the contents of a spreadsheet cell (such as Seyler's button gadget, which when clicked updates the value in the cell represented by the gadget, and the meter, which is updated to reflect contents of the cell), the prior art fails to teach the bidirectional linking between the cell and the user interface object attached to the cell. For example, Seyler's meter gadget reflects cell values, but the value displayed by the meter cannot be directly altered by the user.

(June 13, 1996 Notice of Allowance, DETFH0002516-23) (JCCC at Ex. O) (emphasis added)

DET's "no construction" proposal for "linking" as used in the claims would fail to inform the jury of this "bi-directional" requirement of the claimed invention. Google's proposed construction is thus proper.

3. "Displaying said user interface object with a value of said value property of said value property of the given cell object"

Claim Term(s)	Google's Proposed Construction	DET's Proposed Construction
"Displaying said user interface object with a value of said value property corresponding to the value of said value property of the given cell object" Claims: 1.	Automatically updating the value of the value property of said user interface object when the value of the value property of the corresponding cell is changed.	Plain and ordinary meaning. No construction necessary.

Similar to the "linking" terms identified above, the dispute over this term concerns the "bi-directional" linking requirement of the invention. Google's proposal gives the "displaying said user interface…" term the definition that the applicant himself gave the term during prosecution to demonstrate the "bi-directional" nature of his invention over prior art. DET's "plain and ordinary meaning" construction would ignore this aspect of the invention, attempting to cover systems in which the only requirement is that a user interface object and a linked cell display the same value.

The '591 patent applicant explained on numerous occasions during prosecution that the invention involved "bi-directional" linking between a user-created user interface object and one or more cells. In addition to and confirming the above, the applicant specifically stated that the "displaying said user interface object…" term in the claims means "changing the value of said value property in a cell will update the corresponding value of the user interface object." (June 7, 1996 Interview Summary, DETFH0002514 (JCCC at Ex. N) ("Applicant confirmed that in claim 1 (lines 18-20), the 'displaying said user interface property…' step is to be interpreted to mean that changing the value of "said value property" in a cell will update the value of the user interface object's 'value property.'). The patentee provided this express definition to demonstrate that according to the invention, not only are changes to the user interface object reflected in the cell, but conversely changes to the cell are reflected in the user interface object. The examiner cited this applicant-supplied definition in the Notice of Allowance granting the patent.

> In accordance with applicant's argument at page 9, and in accordance with the interview with David Slone on 6/7/96, the 'displaying said user interface object' limitation at lines 18-20 of claim l is to be interpreted to mean that a change in the cell value is reflected in the user interface object. Thus, the interpretation of claim 1 as a whole shall be construed to mean that a bidirectional linking between the cell value property and the user interface object value property exists such that when one object's value property is changed, the other's value property will be updated.

(June 13, 1996 Notice of Allowance (DETFH0002516-23), at p. 2) (JCCC at Ex. O) The Court should adopt Google's (and the patentee's) construction of this term just as the patent examiner adopted it to grant the patent. Allowing a "no construction" construction would leave otherwise broad claim language to cover any situation in which a user interface object and cells are in any way associated, and omits a key feature (as explained by the inventor himself).

4. “End-user input that effects a change in the value of said value property of said user interface object”

Claim Term(s)	Google’s Proposed Construction	DET’s Proposed Construction
End-user input that effects a change in the value of said value property of said user interface object Claims: 1. End-user input that specifies a change in the value of the display attribute property of said additional user interface object Claims: 3.	“End-user input” means Input by a person who runs a completed application to perform tasks.	Input by a person who is running a custom application operative in an electronic spreadsheet to perform tasks.

For the term, “end-user,” Google proposes a construction straight from the specification:

> For purposes of the following description, it is necessary to differentiate between a developer user ("developer") and an end-user (“user”). A developer, as used herein, refers to the person who is building a spreadsheet application. A user, on the other hand, refers to the person who runs a completed application, that is, one who is interested in using the application to perform tasks.

(’591 Patent, 25:59-26:53.) Google’s proposed construction of “end-user” is taken directly from the specification’s only explanation of the distinction between a developer user and an end-user. Google respectfully requests that the Court adopt its proposed construction.

VI. **CONCLUSION**

For the foregoing reasons, Google respectfully requests that the Court adopt its proposed constructions for the disputed claim terms and phrases set forth herein.

OF COUNSEL:

Jonathan K. Waldrop
Darcy L. Jones
Robert P. Watkins III
Marcus A. Barber
John Downing
Heather S. Kim
KASOWITZ, BENSON, TORRES & FRIEDMAN LLP
333 Twin Dolphin Drive, Suite 200
Redwood Shores, California 94065
(650) 453-5170

Jeffrey J. Toney
KASOWITZ, BENSON, TORRES & FRIEDMAN LLP
1349 West Peachtree Street N.W., Suite 1500
Atlanta, Georgia 30309
(404) 260-6080

Dated: November 3, 2015

/s/ Jason J. Rawnsley
Frederick L. Cottrell, III (#2555)
Jason J. Rawnsley (#5379)
RICHARDS, LAYTON & FINGER, P.A.
920 N. King Street
Wilmington, Delaware 19801
(302) 651-7700
cottrell@rlf.com
rawnsley@rlf.com

Attorneys for Defendant
GOOGLE INC.

CERTIFICATE OF SERVICE

I HEREBY CERTIFY that on November 3, 2015, I caused the foregoing to be electronically filed with the Clerk of Court using CM/ECF, which will send notification of such filing to all registered participants, and have sent true and correct copies by electronic mail to the following:

Brian E. Farnan
Michael J. Farnan
FARNAN LLP
919 North Market Street
12th Floor
Wilmington, Delaware 19801
(302) 777-0300
bfarnan@farnanlaw.com
mfarnan@farnanlaw.com

Amir Alavi
Demetrios Anaipakos
Brian E. Simmons
Alisa A. Lipski
Jamie Aycock
Benjamin Foster
Kenneth Young
AHMAD, ZAVITSANOS, ANAIPAKOS,
ALAVI & MENSING, P.C.
1221 McKinney Street, Suite 3460
Houston, Texas 77010
(713) 655-1101
aalavi@azalaw.com
danaipakos@azalaw.com
bsimmons@azalaw.com
alipski@azalaw.com
jamieaycock@azalaw.com
bfoster@azalaw.com
kyoung@azalaw.com

/s/ Jason J. Rawnsley
Jason J. Rawnsley (#5379)
rawnsley@rlf.com